

Практическое задание №14 (Наследование. Динамический полиморфизм. Абстрактные классы)

Создать DLL-библиотеку классов **Pz14** (согласно своему варианту) и использующий её проект **Pz14.Tests**. Тестирование методов классов осуществлять с помощью **gtest**. Реализовать иерархию классов с **абстрактным базовым классом (интерфейсом)** согласно варианту (например, **IBase**). Во всех классах реализовать необходимые поля, конструкторы, методы получения значений полей и методы установки значений полей, а также организовать перегрузку нужных операторов. Конструкторы и методы обязательно должны проверять параметры на допустимость; в случае неправильных данных генерировать исключение или возвращать код ошибки.

Все классы иерархии дополнить **статическими** методами **Load**, считывающими информацию из потока ввода, создающими объект соответствующего класса и возвращающими указатель на созданный объект в виде указателя на абстрактный базовый класс (интерфейс):

```
static IBase * Load(std::ifstream& stream);
```

Для возможности сериализации (записи в поток) объектов созданной иерархии классов использовать **порождающий шаблон “Абстрактная фабрика”**, в которой создать реестр указателей на функции, формирующие (в результате чтения из потока) объекты – потомки **абстрактного базового класса (интерфейса)**. Методы Load чтения из потока объекта каждого класса иерархии регистрируются в этом реестре. Реестр организовать в виде контейнера **map** стандартной библиотеки шаблонов STL (см. раздел 4 и Листинг П14.5 в файле «Материалы к Практическому заданию №14») с элементами **ключ** (имя класса) типа **std::string** – **значение** (указатель на функцию – статический метод Load класса).

Сериализацию (запись в поток) объектов созданной иерархии классов осуществлять с помощью специальной функции **void Save()**.

Создать текстовый файл (например, **objects.txt**), содержащий информацию об объектах различных классов иерархии (в одном файле в произвольном порядке построчно должна храниться информация об объектах разных классов иерархии – **имя класса** в виде строки и **значения полей объектов**), например для классов **Teacher** и **Student**:

Teacher Smolyaninov Vladimir Yuryevich 1978 K-731 МП РПСПО ЯП

Teacher Bolovtsov Sergey Vladimirovich 1991 K-733 АСВТ МП ЯП

Student Egorov Ivan Ilyich 2000 7321 5 4 5 5

Teacher Petukhova Nadezhda Nikolaevna 1900 K-733 МП ЯП

Student Trunov Oleg Petrovich 2000 7322 3 4 3 4

Student Sidorov Denis Vasilyevich 2000 7321 4 5 3 5

Teacher Zhelonkin Aleksey Vladimirovich 1995 K-733 МП ЯП

и т.д.

Десериализацию (считывание из потока) объектов созданной иерархии классов осуществлять с помощью специальной функции **void Load()**.

В тестах продемонстрировать и проконтролировать работу всех функций и методов всех классов иерархии.

Вариант 1

Создать абстрактный класс – интерфейс **IDetail** (деталь) с виртуальным методом вычисления стоимости детали. Создать производные классы **Box** (короб) и **Handle** (ручка) с членами: материал детали (определяющий стоимость единицы веса), вес, категория сложности изготовления.

Вариант 2

Создать абстрактный класс – интерфейс **IFlash** (тело) с виртуальными методами вычисления площади поверхности и объема. Создать производные классы **Conus** (конус) и **Cilinder** (цилиндр).

Вариант 3

Создать абстрактный класс **IPair** (пара целых чисел) с виртуализацией внешних функций вставки в поток и извлечения из потока (operator << и operator >>). Создать производные классы **Complex** (комплексное число) и **Fraction** (дробь, заданная числителем и знаменателем) и применить к объектам этих классов функции вставки в поток и извлечения из потока.

Вариант 4

Создать абстрактный класс **IChangearray** (изменяемый массив) с виртуальными методами сортировки и поэлементной обработки массива. Создать производные классы **Bubble** и **Choice**. В классе **Bubble** сортировка реализуется методом пузырька, а поэлементная обработка состоит в извлечении квадратного корня. В классе **Choice** сортировка реализуется методом выбора, а поэлементная обработка – вычисление натурального логарифма.

Вариант 5

Создать абстрактный класс – интерфейс **ISeries** (прогрессия) с виртуальными методами вычисления j -го элемента прогрессии и суммы n членов прогрессии. Создать производные классы **Linear** (арифметическая) и **Exponential** (геометрическая). Арифметическая прогрессия $a_j = a_0 + jd, j = 0, 1, 2, \dots$ Сумма арифметической прогрессии: $s_n = (n+1)(a_0 + a_n)/2$. Геометрическая прогрессия: $a_j = a_0 r^j, j = 0, 1, 2, \dots$ Сумма геометрической прогрессии: $s_n = (a_0 - a_0 r)/(1 - r)$.

Вариант 6

Создать абстрактный класс – интерфейс **IFunction** (функция) с виртуальными методами вычисления значения функции $y = f(x)$ в заданной точке x и вывода результата на экран. Создать производные классы **Ellipse** (эллипс) и **Hyperbola** (гипербола).

Уравнение эллипса $\frac{x^2}{a^2} + \frac{y^2}{b^2} = 1$; гиперболы: $\frac{x^2}{a^2} - \frac{y^2}{b^2} = 1$.

Вариант 7

Создать абстрактный класс **IPair** (пара целых чисел) с виртуализацией внешних функций вставки в поток и извлечения из потока (operator << и operator >>). Создать производные классы **Money** (рубли и копейки) и **Time** (часы и минуты) и применить к объектам этих классов функции вставки в поток и извлечения из потока.

Вариант 8

Создать абстрактный класс – интерфейс **ICalculation** (вычисление) с виртуальными методами вычисления значения функции $y = f(x)$ в заданной точке x и вывода результата на экран. Создать производные классы **Sinus** (синус $y = \sin(ax + b)$) и **Parabola** (парабола $y = ax^2 + bx + c$).

Вариант 9

Создать абстрактный класс – интерфейс **IPay** (оплата) с виртуальным методом вычисления стоимости оплаты. Создать производные классы **Flat** (квартира) и **Phone** (телефон).

Вариант 10

Создать абстрактный класс – интерфейс **IProcessing** (обработка) с виртуальным методом обработки объекта. Создать производные классы **Array** (массив чисел) и **String** (строка символов). В классе **Array** обработка реализует домножение каждого элемента массива на среднее, а в классе **String** – замену всех букв на заглавные.

Вариант 11

Создать абстрактный класс – интерфейс **IFigure** (плоская фигура) с виртуальными методами вычисления площади и периметра. Создать производные классы **Rectangle** (прямоугольник), **Circle** (круг), **Trapezium** (трапеция).

Вариант 12

Создать абстрактный класс – интерфейс **ICurrency** (валюта) с виртуальными методами для работы с денежными суммами. Создать производные классы **Dollar** (доллар) и **Euro** (евро).

Вариант 13

Создать абстрактный класс **ITriad** с тремя целочисленными полями и виртуальными методами увеличения на 1 значений полей класса. Создать производные класс **Time** (время) с полями час, минута, секунда и класс **Date** (дата) с полями год, месяц, день.

Вариант 14

Создать абстрактный класс – интерфейс **IModul** с виртуальным методом вычисления модуля. Создать производные классы **Complex** (комплексное число), **Fraction** (дробь). Модуль комплексного числа вычислять как корень из суммы квадратов действительной и мнимой частей; модуль дроби вычислять как корень квадратный из суммы квадратов числителя и знаменателя.

Вариант 15

Создать абстрактный класс – интерфейс **ITriangle** (треугольник) с виртуальными методами вычисления площади и периметра треугольника. Создать производные классы прямоугольный треугольник, равнобедренный треугольник, равносторонний треугольник.

Вариант 16

Создать абстрактный класс – интерфейс **INorm** с виртуальными методами вычисления нормы и модуля. Создать производные классы **Complex**, **Vector3D**. Модуль для комплексного числа вычисляется как корень из суммы квадратов действительной и мнимой частей; норма для комплексных чисел вычисляется как модуль в квадрате. Модуль вектора вычисляется как корень квадратный из суммы квадратов координат; норма вектора вычисляется как максимальное из абсолютных значений координат.

Вариант 17

Создать абстрактный класс – интерфейс **IRoot** (корень уравнения) с виртуальным методом вычисления корней. Создать производные классы **Linear** (линейное уравнение) и **Square** (квадратное уравнение).

Вариант 18

Создать абстрактный класс – интерфейс **IBody** (тело) с виртуальными методами вычисления площади поверхности и объёма. Создать производные классы **Parallelepiped** (параллелепипед) и **Ball** (шар).